1.while(sum(sum(isinf(optTravelTime)))~=numnodes && numTransfer<=maxTransfer)

why two sum not only one.

A:

All cells of optTravelTime are inf at first.

Once the search is finished optTravelTime will only have inf in diagonal.

isinf(optTravelTime) returns the matrix where diagonal cells are 1 and others are 0.

Function sum() sums up the matrix in one directional.

If optTravelTime is n\*n matrix, sum(isinf(optTravelTime)) returns n\*1 matrix.

And sum(sum(isinf(optTravelTime)) returns integer value which is the sum of all cells in optTravelMatrix.

2. based on what optTravelTime get update?

Each cell of optTravelTime means minimum time between two nodes where row is the index of start node and column is the index of end node.

So we should update the cells of optTravelTime when the time between two nodes is smaller than the value of cell.

3. in choosing the isolated nodes =>1. isonodes = find(sum(isinf(travelTimeMatrix),2)==numnodes-2); why numnodes -2

A: For isolated nodes, it has two non –inf values for the time values.

The value for the linked node and self as following.

0 inf inf 1 inf inf… inf

4. in finding the path =>

route = path((i-1)\*maxroutelen+1:i\*maxroutelen+1

here If maxrouteelen=7 and path\_len=15 this will be path(0:8) not oversize > maxrouteelen

A:

No this will be path(1:8) and not oversize maxroutelen.

5. numTranOfBest(nodef,nodel) = numTranOfBest(nodef,nodestop)+numTranOfBest(nodel,nodestop)+1;

numTranOfBest(nodel,nodef) = numTranOfBest(nodef,nodel);

the numTranOfBest is defined to contains inf values how does it get suddenly int values?

A: numTranOfBest is defined to contains inf values at first.

Cells of numTranOfBest means the number of transfers between two nodes where row is the index of start node and column is the index of end node.

Inf means that the path there is no path between two nodes.

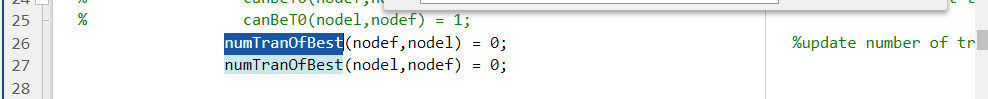
We searched the paths for all pairs of two nodes and once the path is found and the time is smaller than old one, we update the cell of optTravelTime matrix and numTranOfBest.

For example, at first we iterate the pairs in direct transfer.

Since all the cells of optTravelTime is all inf, they are all replaced at this stage.

Of course for the pairs which are linked directly.

At the same time, we should update the cells of numTranOfBest as 0 as following.



Cells of numTransferOfBest which remains inf yet means that there is no direct path for those pairs of nodes.

Next, we search the paths between pairs of nodes which are linked one transfer and if the time including transfer time is smaller than before, we update the optTravelTime and numTransferOfBest.

Cells of numTransferOfBest which remains inf yet means that there is no path which has length equal or smaller than 1 transfer for those pairs of nodes.

6. for pop\_i = 1:population

fitDistribution(pop\_i+1) = fitDistribution(pop\_i)+fitMatrix(pop\_i);

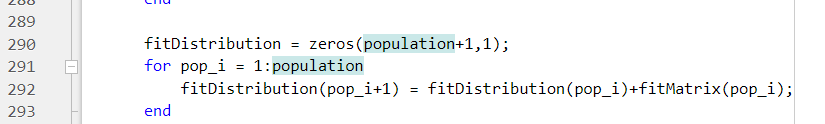
end

fitSum = fitDistribution(end);

fitDistribution = fitDistribution/fitSum;

what does fitDistribution(pop\_i+1) represent and why starts from 2 not 1. and last line will divvectorctre over vector right?

A:



fitDistribution represents the probability distribution for populations and fitDistribution(1) should be zero.

Last line is needed for normalization.

As u know, the probability should be in the range of [0,1].

7.slection

for pop\_i = 2:population

if(rand<1-crossover\_prob) %eleitism percentage 20%,

parent = select\_individual(fitDistribution);

while(~isempty(find(containedInd==parent, 1)))

parent = select\_individual(fitDistribution);

end

-why from i=2

-what is containedInd is it group or individual and

-why we repeat this twice parent = select\_individual(fitDistribution);

- what is fitDistribution

A:

- It is since we save the best solution as the population for next generation and it is saved in parent(1).

-We use eleitism percentage to save some individuals originally in the next generation.

Here, we select the individuals considering the fitness function and the better individuals will have higher percentage to be saved.

However we should avoid to select the same individuals since it will make the solution to converge to the local optimization points and affect the performance of ga.

So we save the selected individuals already in the containedInd and check the selected individual is contained in it whenever we select the individual.

- while(~isempty(find(containedInd==parent, 1)))

parent = select\_individual(fitDistribution);

end

no we select one individual in eleitism.

You should check the while statement.

Pls refer the above answer.

-fitDistribution is described in the above solution.

8. mutation

mutatedIndividual{route\_ind} = [route(1:nodef\_ind-1) path route(nodel\_ind+1:end)];

why not ind and ind+1

A: route(nodef\_ind) is randomly selected first node and route(nodel\_ind) is randomly selected end node.

Path is the shortest path between first node and we replace the path in the route as this.

Here the start node of Path is same as route(nodef\_ind) and end node of Path(nodel\_ind) is same as route(nodel\_ind).

So to avoid repeat of these nodes, we use ind+1 not ind.